Domain-Specific Sentiment Lexicons Induced from Labeled Documents

Fabio Tecco
Politecnico di Torino
285569@studenti.polito.it

Salvatore Junior Curello
Politecnico di Torino
s268066@studenti.polito.it

Alessio Mongoli
Politecnico di Torino
s267501@studenti.polito.it

***Abstract*—Sentiment lexicon is an important tool for identifying the sentiment polarity of words and texts. How to automatically construct sentiment lexicons has become a research topic in the field of sentiment analysis and opinion mining. In this paper we extend the work made by *de Melo et al.* [1] based on Domain-Specific Sentiment Lexicons Induced from Labeled Documents. A word's sentiment depends on the domain in which it is used. In our framework, we rely on labeled documents for a set of different domains of the Amazon Dataset Reviews in order to induce seed data for each of the domains using simple linear predictors. Then, we use deep neural models to learn sentiment intensity scores for larger vocabulary. Starting from this approach, we proposed some extensions: (1) we implemented different negation detection techniques (2) we test our model on two different labeled datasets (3) we perform a further analysis on the changing polarity scores over different range of prices.**

I. Problem Statement

Sentiment analysis or opinion mining is the computational study of people's opinions, sentiments, attitudes, and emotions expressed in written language. It is one of the most active research areas in natural language processing and text mining in recent years. In this paper, we exploit sentiment lexicons, an unsupervised technique that can be run without the need of any labeled training data. A sentiment lexicon is a resource that, for a given word *w*, provides a label that describe its overall sentiment polarity. It can provide a predefined label in $\{$*positive, negative, neutral*$\}$ or just $\{$*positive, negative*$\}$. In some other cases, we can get interesting information about the intensity score of a certain word thanks to the dynamic property of a sentiment lexicon. Adjectives like *good*, *great*, and *excellent* are similar in meaning, but differ in intensity. A ranking process of this information could be crucial in order to better capture different intensities of an emotion, and is hence very useful for humans when judging product reviews [2]. These intensity scores can be represented by a value in the range [-1,1], where -1 denote the most negative sentiment polarity and +1 denote the most positive score.
In particular, depending on the different contexts, we can obtain different sentiment polarity for the same words. However, there are some lexicons sentiment problems: (1) they are usually based on domain-independent notions of sentiment and (2) they are typically manually created, thus having limited coverage polarity.

We use pretrained word vector representations such that we can find sentiment and semantic information, so we can observe that a word like *”delicate”* is positive in some domains and negative in others, and we can also quantify how positive or negative it tends to be in each domain.

In our method, processing a domain-specific corpus $\mathcal{D}$ composed by n documents, we obtain as $\mathcal{X}$ the set of all possible documents generated with the vocabulary $\mathcal{V}$ and $\mathcal{Y}$ as the set of all associated labels. We use SVM classification to predict the polarity score for each word in $\mathcal{V}$. Then we infer a score ϕ for each word with embedding.

II. Methodology

Same as *de Melo et al.* [1] we propose an approach based on two steps. First, we use a set of labeled data to train a linear predictor(SVM) to forecast the score of $w \in V$, also called seed data. Then, we use this seed data to train a deep neural regression network in order to predict the scores to a larger vocabulary.

A. Preprocessing

Given n domain-specific document sets D_i labeled with sentiment polarity labels $Y_i = \{$*positive, negative, neutral*$\}$, we filter out the $neutral$ documents, and we learn n corresponding linear binary classification models using bag-of-words features. Then, each word in the vocabulary is assigned a domain-specific sentiment polarity scores(depending on the reference domain), by consulting the linear coefficients for the respective word.

Basically, our preprocessing phase is divided in some steps. Firstly, we tokenize each review, we lowercase all the tokens, then we apply a negation detection process. We decided to handle the negated words using two different strategies:

- Adding artificial words, two methods are proposed:
  - *Normal Negation Detection*: same as *de Melo et al.* [1] for each D_i we consider a set of features $F_i = V_i \cup \{\bar{w_j} \mid w_j \in V_i\}$ where V_i is the term vocabulary of D_i and $\bar{w_j}$ indicates a denoted version of the word w_j. In particular, we consider as negated feature every word $\bar{w_j}$ preceded by a *”not”* thus we create a new feature $\bar{w_j}$ which will be composed of *”NEG_$\bar{w_j}$”*. For instance, considering a review *”I do not like the quality of this product”* the preprocessed version will be *”I do NEG_like the quality of this product”*. This method is quite fast but in some cases

is not so efficient (e.g. *"I do not NEG_really like the quality of this product"*).

  - *All Words Negation Detection*: same as *Pang et al.* (2002) [3] we added the tag *"NEG_"* to every word between the negation word *"not"* and the first punctuation mark following the negation word. For instance, the output of the review *"I do not like the quality of this product"* will be *"I do NEG_like NEG_the NEG_quality NEG_of NEG_this NEG_product"*. Doing so, we can change the sentiment of the sub-phrase and also capture potentially important contextual effects of negations: *"good"* and *"not very good"* indicate opposite sentiment orientations. However, this is not only linguistically "inaccurate" but also increases the feature space with more sparse features since the majority of words will only be negated once or twice in a corpus. Still, the use of this technique will bring some improvements in terms of accuracy, as will be discussed in the next section.
- No Negation handling: no negation detection technique is applied.

Finally, it is important to point out that the negation detection process is fed with a decontracted version of each review. A function implemented with the use of regular expressions is applied in order to maximize the effect of the negation detection function. For instance, the review *"I don't like the quality of this product"* will be transformed in *"I do not like the quality of this product"*. By doing so, we do not lose the important information that the negation word *"not"* can bring to the analysis.

B. Seed Data Induction

From the preprocessing phase, we obtain the Bag of Words used to train a linear predictor in order to compose the seed data $\{(w_j, p_{ij}) \mid w_j \in V_i\}$. To achieve this we perform some steps:

- We disregard words with frequency lower than a given threshold f_{min} since the higher is the frequency of a word the more is accurate the results.
- We disregard the negated features as in [1], since their frequency tends to be too low to provide a reliable complementary signal, except when we run the experiment with *All_Words_Negation*. Instead to discard the negated feature ($f \geq f_{min}$), we try to combine the polarity of the NEG_w_j with the polarity of w_j using the mean: $\frac{(w_j + NEG\_w_j)}{2}$
- Finally, using *GloVe CommonCrawl* [4] word vectors, we obtain the representation of our vocabulary. The seed data is represented by:

$$T_i = \left\{(w_j, p_{ij}) \mid w_j \in V_i, \sum_{(x,y)\in D_i} f\left(x, w_j\right) \geq f_{min}\right\}$$

where $f(x, w_j)$ denotes the term frequency of the word w_j in document x, p_{ij} is the polarity score of the word w_j and f_{min} is a predefined minimal training corpus frequency threshold.

C. Neural Vector-based Expansion

Here we try to predict the domain specific sentiment polarity for words that are not present in the initial dataset. For this purpose, we train deep neural regression networks model $\phi(v_w)$ based on word vector given by the GloVe CommonCrawl embeddings. After training we use the same model ϕ to predict the sentiment polarity score of words that not appear on the original data T_i. The scores predicted are influenced by the domain of the original data: for instance, when talking about music, the word *"hot"* tends to be positive while when talking about a smartphone, often become more negative. For this task we use the same network as [1], and also the same hyperparameters.

III. Experiments

In this section, we describe the experiments that we conducted. The first experiment is related to Domain Adaptation, which try to apply an model trained in a "source domain" in order to make prediction on a "target domain". In the other experiment, instead, we try to expand the data using the metadata available on the official site, to be able to extract more information and identifying more significant insights. As hardware we used Colab GPU, we trained a deep neural regression model for 100 epochs, we relied on a batch size of 32, dropout rate of 20%, and Adam optimization with an initial learning rate of 0.001 and early stopping.

A. Testing on different domain datasets

The goal of this experiment is to verify how our model can be applied on various domains, represented by different datasets. Indeed, we use three different datasets for training and two for testing. Each training dataset is made up of Amazon reviews from May 1996 to July 2014. In particular:

- *"Movies and TV"* contains 4,607,047 reviews of movies and series
- *"App for Android"* contains 2,638,173 reviews of android apps
- *"Grocery and Gourmet Food"* contains 1,297,156 reviews of groceries

Once obtained the polarity score for each word in GloVe embeddings, we make predictions on unseen reviews from two different datasets:

- the IMDB dataset consisting of 25000 positive and 25000 negative reviews
- the Twitter Airlines dataset consisting of 2300 positive and 2300 negative tweets which express the feelings of travelers in February 2015. The dataset has been preprocessed removing mentions, tags and emoticons.

Each training dataset is managed with four different technique of negations handling: *normal*, *all words*, *all word with negated features* and *no negation*, as explained in section II. After discarding the reviews with a score equal to 3(neutral),

Training	Movies and TV		App for Android		Grocery and Gourmet Food	
Negation Type / Test dataset	**imdb**	**twitter Airlines**	**imdb**	**twitter Airlines**	**imdb**	**twitter Airlines**
Normal	0.88598	0.78935	0.82094	0.78804	0.80356	0.78607
All words	0.88554	0.78826	0.83162	0.79609	0.80842	0.77890
All words with negated features	0.86986	0.76217	0.81000	0.75152	0.78730	0.74673
No negation	0.88470	0.78413	0.82628	0.78607	0.80178	0.77867

TABLE I: Accuracy of the model on the test dataset after being trained on the training dataset with different negation handling technique

Training dataset	Movies and TV				App for Android				Grocery and Gourmet Food			
Phases / Negation type	**normal**	**all words**	**all words neg**	**no negation**	**normal**	**all words**	**all words neg**	**no negation**	**normal**	**all words**	**all words neg**	**no negation**
Preprocessing	687	1619	1621	1311	177	380	417	396	130	268	272	252
TrainSVM	3153	3661	3669	3211	1052	1103	1072	1179	569	513	515	558
GloVe	912	1250	1290	1027	241	246	261	206	208	270	242	197
Training	403	554	412	460	147	147	154	157	155	148	177	151
Predict	54	77	62	64	85	85	92	91	86	86	86	84
TOTAL	5209	7161	7054	6073	1702	1961	1996	2029	1148	1285	1292	1242
Seconds for 10k review	11.30	15.54	15.31	13.18	6.45	7.43	7.57	7.69	8.85	9.90	9.96	9.57

TABLE II: Computational time of the different phase of the pipeline expressed in seconds

we only keep the words that occur more than 500 times. Then we also filter the common words to GloVe, in order to obtain the GloVe embedding associated with the word. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. We compute the polarity of each review as following:

$$f(x) = \frac{1}{|x|} \sum_{i=0}^{|x|} \phi(\vec{v}_{x_i})$$

where $|x|$ denotes the document length and x_i the i-th word in x and $\phi(\vec{v}_{x_i})$ is the neural prediction score. Then, we predict the polarity by setting the average of all such prediction scores in the corpus as a binary threshold.

In TABLE I we reported the results of the various combinations of training dataset, test dataset and negation type. The metric used is the accuracy which is the fraction of predictions that our model got right. The model trained on *"Movies and TV"* has very similar accuracies to changing the type of negation for the same type of test dataset. Also, as expected, the model gives more accurate predictions for the "IMDb" test dataset, because the two datasets have a similar distribution of data, in fact both training and test deal the same topics. To underline this observation we choose to use as second test dataset one which is not mainly composed of reviews, in fact the Twitter Airlines dataset is a set of tweets("opinions"). So there is a discrepancy across domain distributions, and naively applying the trained model on this test dataset can cause lower performance than the "IMDb" dataset, but still quite acceptable. As expected, *"Grocery and Gourmet Food"* has the lower performance both because the domain is totally different and the size of the training dataset is smaller. As regards the times, we can observe in TABLE II that most of the time is spent in the training phase of the SVM model and in the preprocessing phase of the reviews, because the creation of the seed data need a more computational effort.

B. How the price can influence the rating

We conduct an experiment aimed at analyzing how the price can influence the opinion of a reviewer. The goal of this experiment is to show how an increasing price may have a negative effect not only on consumer rating, but consequently also on the sentiment of a review. To do so, we used as dataset *Amazon Musical Instruments* which composed of 500,176 reviews from May 1996 to July 2014. Then, we augmented it using its metadata which contain further information about descriptions, price, sales-rank, brand info, and co-purchasing links of 84,901 Musical Instrument products. In this way, we constructed a new dataset, in which to each review is associated its rate and its price. We conducted three different analyses. We decided to perform an equally partitioning of the whole dataset according to the total number of samples over the difference values of the price. In this way, each portion of the "sub-dataset" contains the same number of reviews. This experiment is conducted using the Normal Negation Detection Technique.

The first analysis is made to show the different number of positive and negative words over the three different price ranges. TABLE III shows some statistics about each portion which can be useful to find some insights on how the price can influence the polarity of a token. In particular, for the low price, we have that the number of negative words is half than the number of positive words. This can be explained by the fact that a lower price always increase the satisfaction of a consumer and at the same time if the price of a certain product is lower the buyer is more likely to accept some defects or a lower quality. For the medium price, as expected, the number of positive and negative words is more balanced. Finally, for the high price, we observe that the number of negative words is twice the number of negative words of the low portion. This could be caused by two reasons (1) an higher average of the length of the reviews as we will discuss later and (2) an increasing in term of price followed by a consequent increase of the quality of a product does not always bring an higher rate from the buyers, because the higher is the price that a buyer is willing to pay, higher is the level of satisfaction that product bought must fulfill. In addition, usually, a buyer who has a negative experience is highly likely to share that experience

Price	MinPrice	MaxPrice	MeanPrice	Num of reviews	Mean Lenght reviews	Num positive words	Num negative words	Total num of words
Low	1	18	9.84	144845	67	911227	439920	1351147
Medium	18	60	33.98	144845	79	751316	652943	1404259
High	60	1000	192.94	144845	106	911637	712253	1623890

TABLE III: Statistics of the Musical Instrument Dataset with the same configuration of the first experiment, using *Normal* negation detection

Negative tokens		
Low Price	**Medium Price**	**High Price**
useless	*worst*	*worst*
poor	*useless*	*waste*
worthless	*horrible*	*returning*
wretched	*disgraceful*	*wastes*
pitiful	*deplorable*	*disappointing*
unproductive	*inhumane*	*mediocre*
ineffectual	*disgusting*	*poorest*
miserable	*dismal*	*miserable*
pathetic	*appalling*	*dismal*
Positive tokens		
Low Price	**Medium Price**	**High Price**
satisfied	*pleasantly*	*pleasantly*
excellent	*brilliant*	*enjoying*
ledger	*agreeably*	*unlikeness*
beautifully	*amazing*	*pleasently*
incredible	*adreline*	*surprised*
pleased	*informative*	*flawless*
superb	*satisfied*	*pleasurably*
wonderful	*delicious*	*superb*
approved	*awesome*	*good*

TABLE IV: List of the most interesting positive and negative tokens

Token	Low Price	Medium Price	High Price
drums	-0.14877	0.06449	0.08565
delicate	0.00905	0.33861	0.31517
package	-0.1867	0.03338	0.05841
delivery	-0.1170	0.13268	0.27444

TABLE V: List of the most interesting positive and negative tokens

by leaving a bad review but on the other hand who has a positive experience is unlikely to spend time to leave a good review.

A second analysis is performed in order to extract the most significant positive and negative words reported in TABLE IV. We can see that the higher is the price the more "linguistically" intensive is the meaning of the word. Particularly interesting is the token *"returning"* that appear in the high price. This shows how an higher price push the buyers to return back a product if they are not totally satisfied. Instead, for the low price as expected the tokens *"unproductive"* or *"ineffectual"* show how a product with a low price reflect the quality. Also, we extracted some significant tokens together with their polarity values reported in TABLE V in order to show the change over the different ranges. In particular, the tokens "package"(from -0.18 to 0.05) and "delivery"(from -0.12 to 0.27) show the different quality of the service offered with the respect to the price. Generally, the lower is the price the lower is the level of attention put on it.

Finally, the third analysis is aimed at inspect how the length of the reviews can change according to the different ranges of price. As already shown in TABLE III, we can observe that the length of the reviews are directly proportional to the increase of the prices. To find some further evidences that support this observation, we performed a deeper analysis also on others categories of the Amazon Dataset. In particular, we also considered *"Baby"*, *"Cell Phones and Accessories"*, *"Tools and home Improvement"* and *"Office Products"*. Figure 1 shows the average length of the reviews according the different range of prices. We can observe that among all of these categories an increase of the price bring an increase of the average length of the reviews. From low price to medium price we have an increasing of 10 words while from medium price to high price the increase is about 26 words. It is interesting to notice that the increase of the number of words for the high price with respect to the low price shows that a buyer is likely to spend time on writing a longer and more detailed review when he/she pays an higher price.

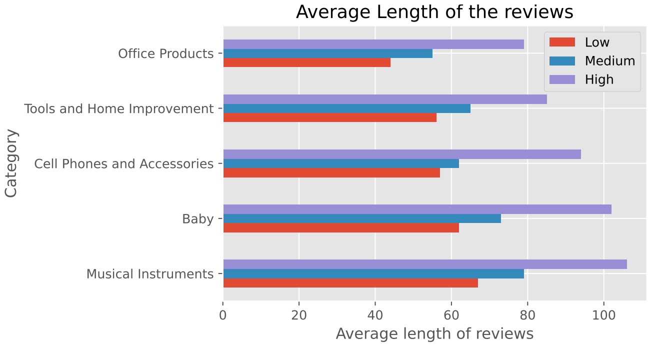

Fig. 1. Average number of words of the reviews over different range of price

IV. Conclusion

In this paper we present that negation detection has an important role in the sentiment lexicon as shown in the experiment 1. We also shown that there are substantial differences between domains, which make domain-specific lexicon an important resource. In order to improve our work is possible to conduct more sophisticated negation detection techniques. Consequently, we can assume that this technique may work better in those domains with contexts that are not too decisive for determining sentiment analysis. One way to extend this work could be to create domain-specific sentiment lexicons that handle emoticons and other techniques, which can capture irony and sarcasm, especially because they can flips the polarity of the text.

References

[1] S. M. Islam, X. Dong, and G. de Melo, “Domain-specific sentiment lexicons induced from labeled documents,” in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 6576–6587, International Committee on Computational Linguistics, Dec. 2020.

[2] G. de Melo and M. Bansal, “Good, great, excellent: Global inference of semantic intensities,” *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 279–290, 2013.

[3] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? sentiment classification using machine learning techniques,” in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pp. 79–86, Association for Computational Linguistics, July 2002.

[4] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.